Fahrenheit Co.

Communities

Project Technical Report

Eric Anderson - Diagrams 1-3
Christopher Bednarz - Pseudo Code Flow Charts
Zoe Chamlee - Specifications and Requirements
James W. Garrett - Standards and Editing
Mason Holley - Executive Summary, Goals, and Objectives
Rishabh Tewari - Timeline and Tasklist

Date of Submission: October 11th, 2021

Executive Summary

Communities is an app which allows people within a local scene to have an easier time interacting with each other on a more personal level. Our program does this by isolating the scope of what the client can see to their local areas. This app consists of a map of the user's local area that will allow them to browse and join local groups and communities. They will be able to navigate the groups they join via a feed that will post all info related to the communities.

Overall Goal & Objectives

Our focus in creating Communities is to provide a platform for local communities to grow and flourish in a healthy environment. Communities will allow users to connect with or create a local scene for all kinds of interests, hobbies, or activities. With each user that joins our platform, we gain a new potential member for one of the many communities. While a user might join for one specific community, we want them to be able to find many communities which fit their tastes all in their local area. With this, we believe our company, Fahrenheit Co. will be able to cultivate a healthy user-base which will allow continuous app growth.

We hope to diversify from other similar applications such as Discord and Reddit by location-locking our communities. Each community will have the tools available to announce and schedule events related to each community which will be easily accessible and known to all members. We will also be working to provide tools to allow the creators or appointed leaders of a community to maintain and moderate the community.

Explanation

Each time the user visits the site, our web app requests the user's location. The user is simultaneously given the option to sign in or register.If the user chooses to give their location information, the user will be shown a list of communities on a map that are within their city's coordinate range. The user can search through communities with a search bar. Once they find a community that they think they will like, if they are signed in they will be able to subscribe and receive notifications whenever the community has a post or event.

If the user would like to create an account, they are brought to a prompt for their information. Once a unique username and password have been entered, they are sent to our server it can be stored and encrypted with salted hashes.

If the user would like to create a community, they can do that through the “Create a Community” button. The user will enter the name, tags, and rules of the community befoie selecting their city. After all of this, the community will then be stored made visible to all users in the local area.

Each community will contain a message board, moderation tools, and an event scheduler. The message board contains a list of posts. Each post will have the text content of the post, the poster’s username, and the poster’s avatar, along with a timestamp of when it was posted. Events will contain a description of an event, an optional location, and a time.

Specifications & Requirements

Our specifications can be split up into our hardware specifications and software specifications which are comprised by our software stack. Our hardware specifications at the moment are minimal, a VPS with 1GB memory, 25GB storage, and a single core with 1TB of bandwidth a month. We currently only need a small unified server to run our webapp at the moment which all team members can deploy to and test with, so these minimal specs will suffice.

For the software stack, we will be utilizing Alpine Linux for the server OS, a very light distribution of Linux suited to our small VPS. For the backend portion of our stack, we intend to use MariaDB for the database, Go’s Revel for backend framework, nginx for the webserver, and OpenStreetMap Nominatim API for location services. MariaDB was chosen as it is essentially an open source version of MySQL, which most of our group is familiar with. Go’s Revel was chosen as Go is an easy language to learn for the portion of our group that does not know it and Revel itself for it’s “full featured” web framework in Go. Nginx was selected simply for being standard fare, ubiquitous, and performant. Finally the Nomination API was chosen since it is simple, high quality, and widely used. For our frontend stack, we’re just using React, since it’s what our frontend developer knows and is very ubiquitous.

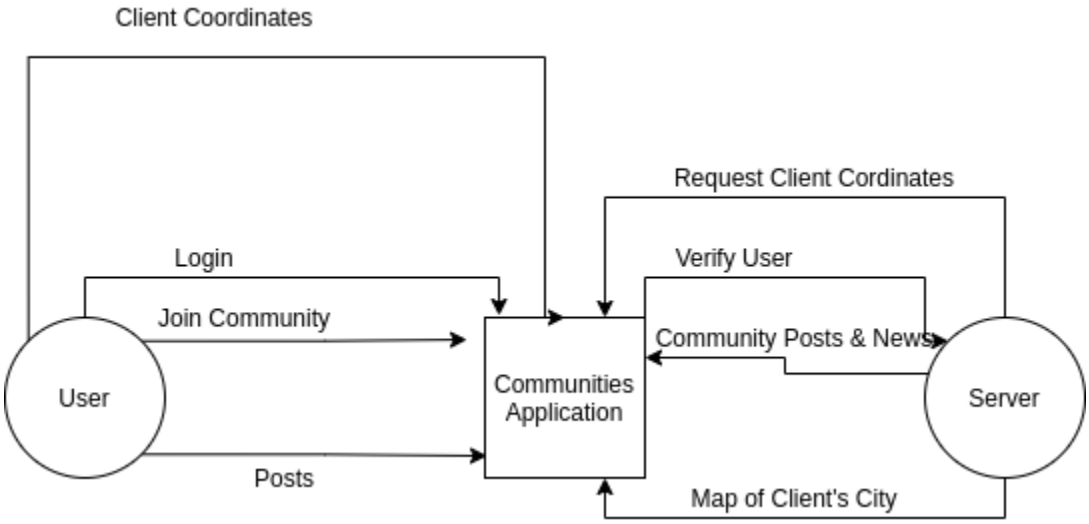
Standards

In so far as standards, we have three main points that we are aiming to hit. While our app may be relatively mundane, we will be dealing potentially very sensitive data given that our program will be using client locational data in order to provide appropriate features. First, we will be taking a look at the International Organization for Standardization or the ISO IEC 20000-1 more specifically concerning our information protection. Currently, our plan is to protect and encrypt our information by making a hash to make the information not readily available. Should this method prove to be not secure enough and we need more, we will look into heavier encryption.

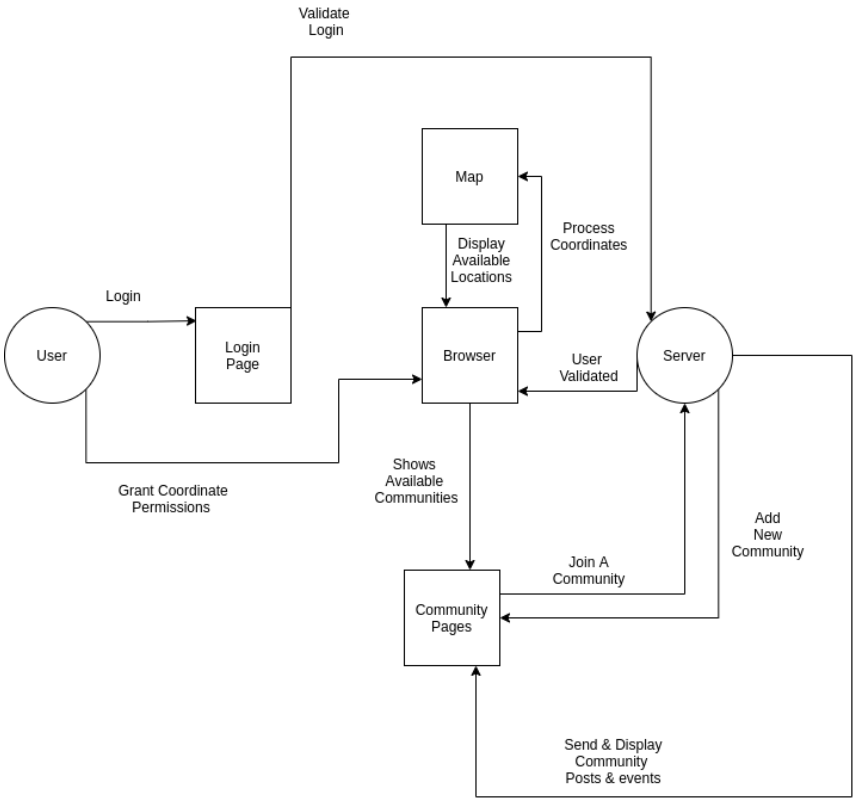
Secondly, our project will be concerned with the General Data Protection Regulation or GDPR. In order to follow this, we will be taking precautions a few ways. The first of which is that we will be asking for client permission before we take any data. Once we have permission, we will want to use it to give them local ads. In order to do this we will be limiting the amount of information that we will be giving to the ads themselves to prevent any problems on that front.

Lastly, this company shall ever be working to adhere to the standard Software Engineering Standards of Ethics. This project in particular is designed to adhere as it is intended to work to the public's interest. Our revenue model allows us to give the app out for free which should further enable local communities to form and thrive. To summarize, our group will be taking precautions and doing preemptive work to keep our project ethical and up to standards.

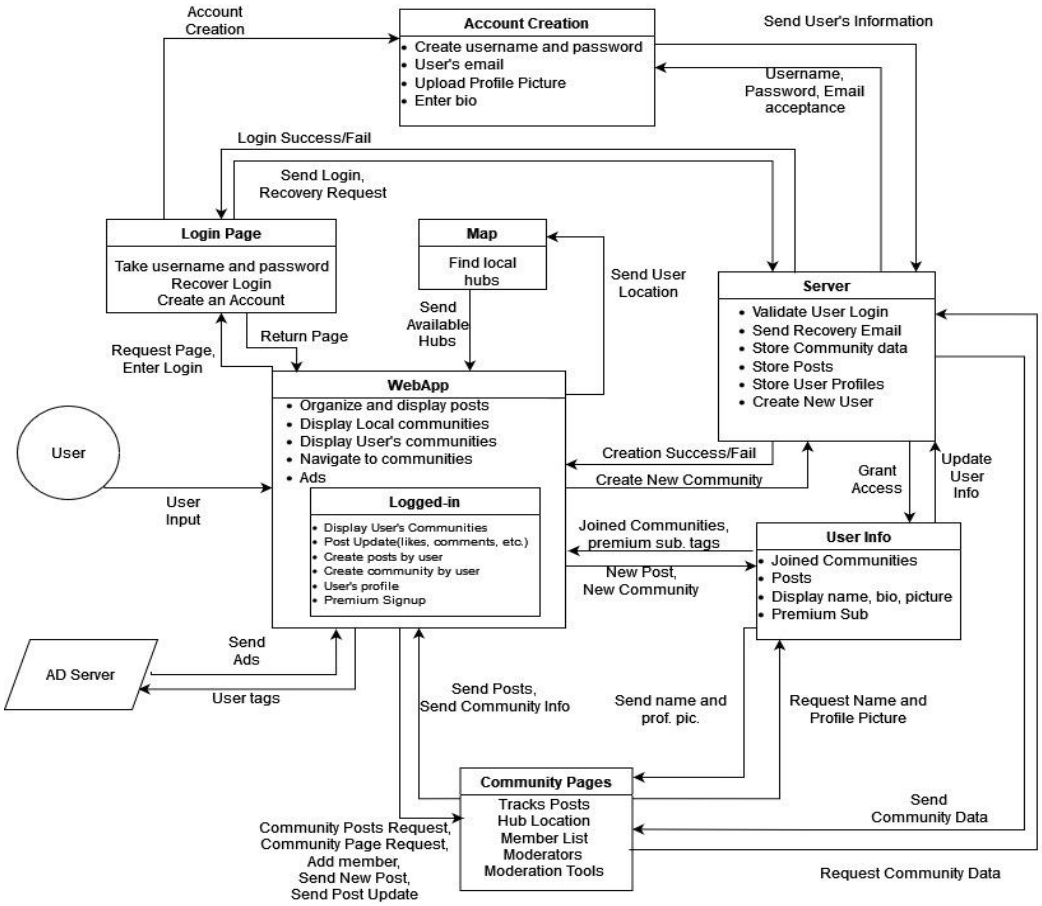
Diagrams



Our level 1 diagram consists of the user, the application, and the server. The user will be able to log in, join communities, make posts, and send his/her coordinates. The server will verify the user, and send the maps, communities & news to the application.



Our level 2 breaks down the application into the login page, browser, map, and community pages. The login page takes in the username and password of the user, which is then validated by the server. After the validation, the user is taken to the browser, which is the webapp for navigating the application consisting of the community pages and the map. The map will process the user's geolocation coordinates and display the user's area. The community pages will display the posts, news, and events from the communities to which the user is subscribed.



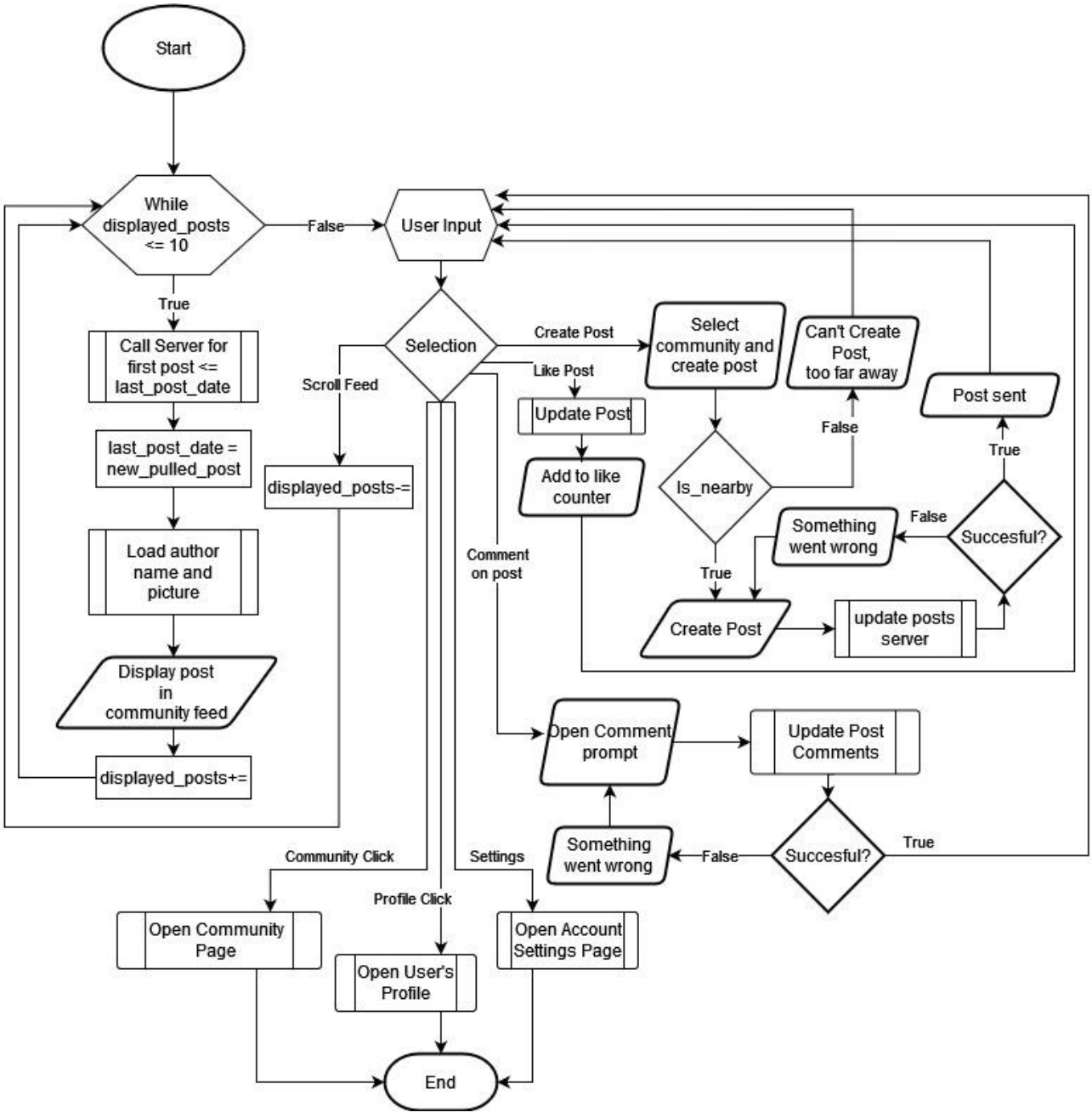
Our level 3 takes a more detailed look into all of the functions and processes planned and breaks down the application even further. There's two levels of the browser, one for non-registered users and one for registered users. The browser for registered users will inherit the functionality of the browser for non registered users. In order to run the ads necessary for revenue, we will have a separate server that is solely responsible for them. In addition, this chart goes deeper into the account creation process where the user is taken to a separate page. This page communicates with the server to determine what is and isn't a valid username and or password. In addition, we will also have user information pages which can be reached through a user's homepage or a Community Post.

Pseudo Code Flowcharts

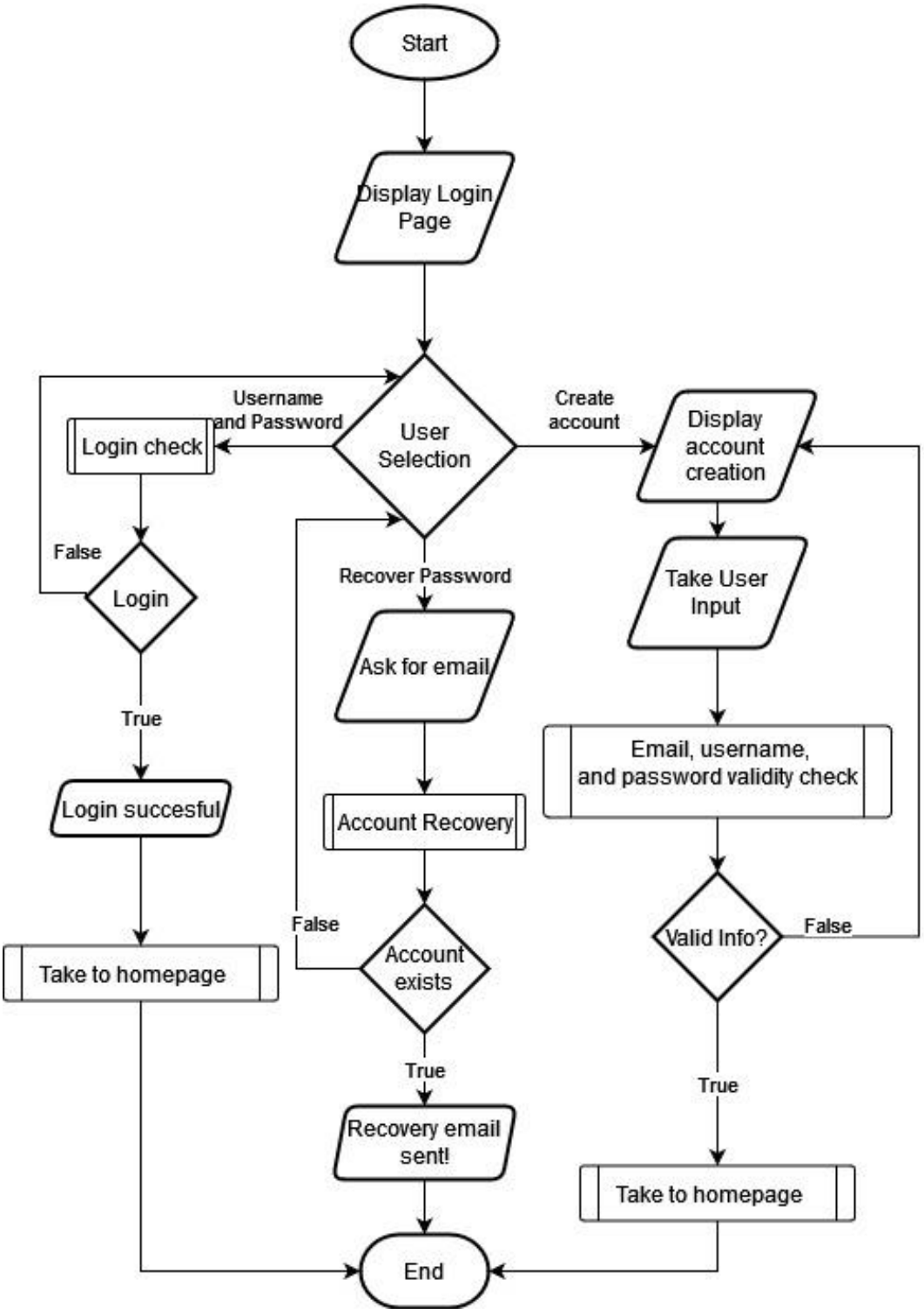
By mapping out data flow and user interaction with the app, we can visualize error handling and prevent the user from performing unwanted actions, as well as pinpoint at which specific point in execution will certain data need to be called and used. A few cases we need to be weary of are: losing connection to the server during communication, user entering invalid data, user being notified of errors, user being notified of invalid actions (such as attempting to

post in a community they're too far away from), and making sure all possible user input is accounted for and properly handled (such as which buttons and menu options they are allowed access to.) You can view just a few examples of this handling below.

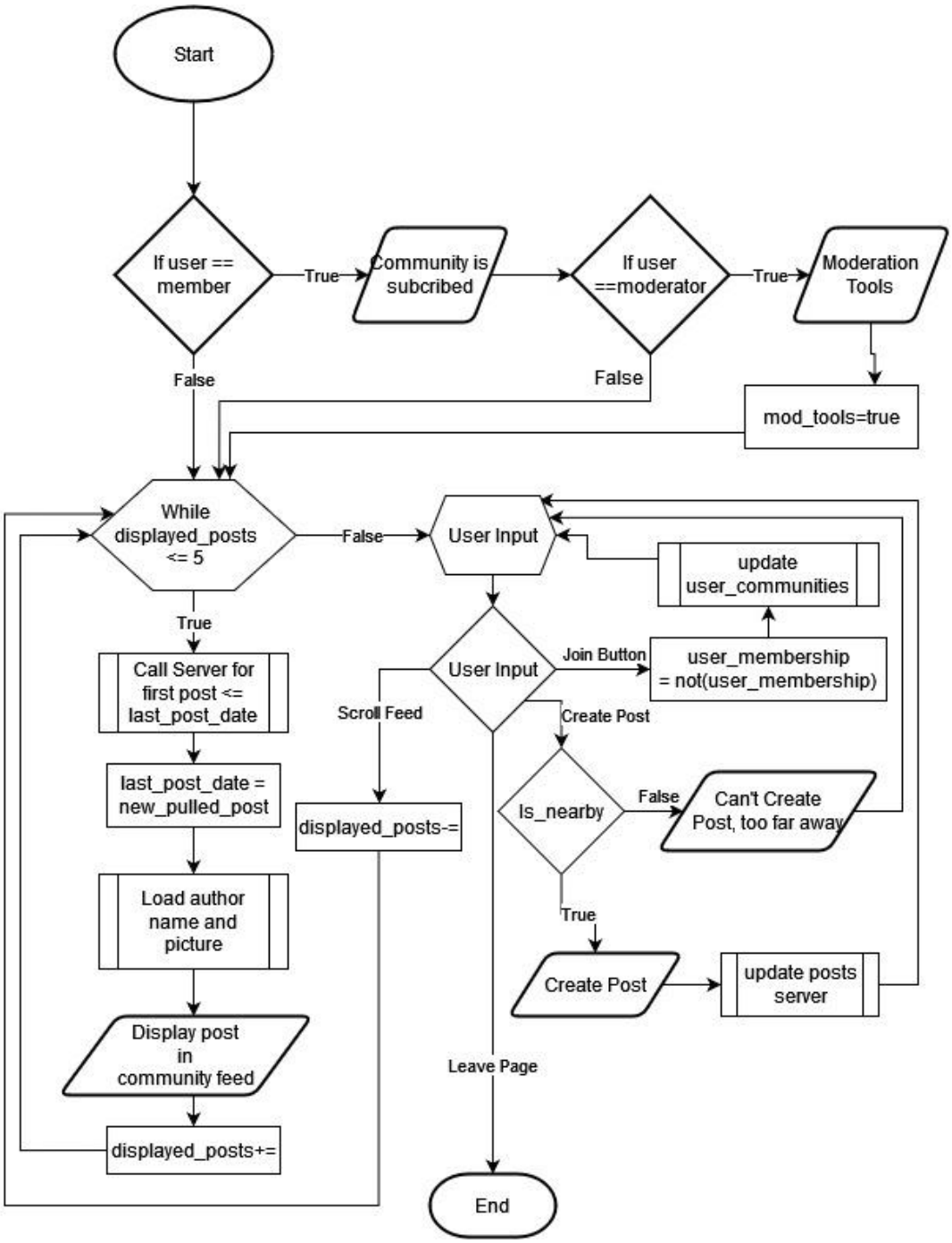
WebApp UI



Login/Account Creation Page

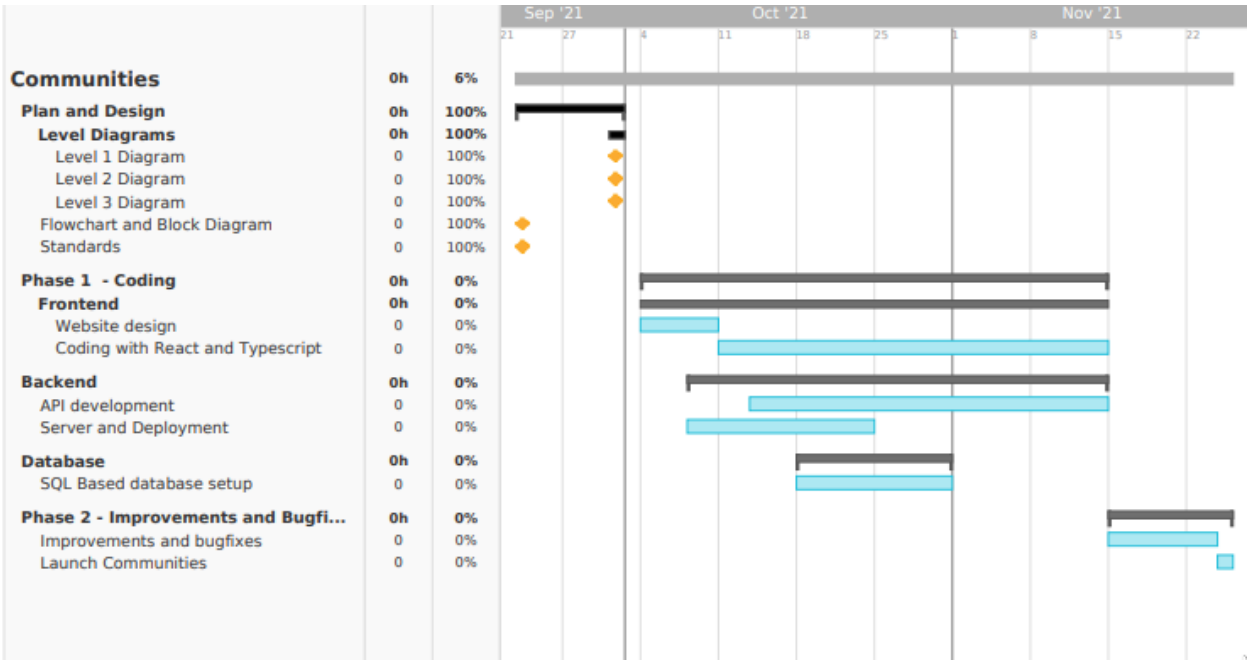


Community Page



Timeline

The timeline for this project has been divided into two phases, with a pre development phase where we dealt with designing and planning our product. Phase 1 of the project deals with the initial coding of the project, where we building towards our MVP (Minimum viable product). Phase 2 of the project deals with proposed improvements and bug fixing. A first draft of the proposed timeline is presented in the Gantt chart below. We have chosen to split up our project in this manner to make sure that we meet minimum standards before ironing out any issues and adding



List of tasks

The responsibilities for each part will be taken on by the following. Will and Christopher will be working to create the databases and working with Zoe to make sure the server is appropriately set up and the data can be referenced. Zoe will be mainly responsible for the deployment and standup of the aforementioned servers. Mason will be responsible for the implementing the Geolocation API and translating the information into workable data. Eric will be working on the framework of the project itself on both the frontend and backend. Rishabh will be working as a fullstack developer with a focus on the Frontend due to his experience.

The list of responsibilities taken by each person are listed below:

Will - Database

Christopher - Database

Eric - Frontend & Backend.

Zoe - Server Deployment & Standup

Mason- Geolocation API Implementation

Rishabh - Frontend & Full Stack

Version Control

We're managing our project using a version control technology called git, and hosting it on an online service called Github. Each iteration of the code will be able to reviewed and monitored by each group member preventing multiple variations of the code. Provided we use proper commenting, we should not have any particular problems maintaining consistency.

References.

CARLILE, LIZ. *Development Online: Making the Most of Social Media.* International Institute for Environment and Development, 2011, www.jstor.org/stable/resrep01460. Accessed 13 Sept. 2021.